

**VŠB - Technická univerzita Ostrava**  
**Fakulta elektrotechniky a informatiky**  
**Katedra 456 - Katedra informatiky**

**Popis komunikace mezi PC se software MATLAB a měřičem výkonu  
Thorlabs PM100 přes RS232**

**Communication Between Thorlabs PM100 and PC using MATLAB  
through RS232**

## **Zadání**

**Téma:** Popis komunikace mezi PC se software MATLAB a měřičem výkonu Thorlabs PM100 přes RS232

**Student:** Pavel Juchelka

**Vedoucí bakalářské práce:** Ing. Jan Skapa, Ph.D.

**Akad. rok:** 2010/2011

**Obor:** 2612R025 - Informatika a výpočetní technika

### **Zásady pro vypracování:**

Cílem práce je realizovat funkční komunikační rozhraní mezi měřičem výkonu Thorlabs PM100 a PC se software MATLAB přes RS232.

1. Stručný popis rozhraní RS232.
2. Popis instrukční sady a syntaxe PM100.
3. Realizace funkčního komunikačního rozhraní mezi PC se software MATLAB a měřičem výkonu Thorlabs PM100.

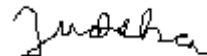
### **Seznam doporučené odborné literatury:**

[1] ZAPLATÍLEK, K.; DOŇAR, B. [i]MATLAB: tvorba uživatelských aplikací.[/i] Vydání 1. Praha, BEN - technická literatura, 2004. 215 s. ISBN 80-7300-133-0.

**Prohlášení studenta:**

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne 29.4.2011



.....  
podpis studenta

**Poděkování**

Tímto děkuji svému vedoucímu bakalářské práce, Ing. Janu Skapovi, Ph.D., za rady, připomínky a náměty. Rovněž mu děkuji za konzultace mimo období výuky.

## **Abstrakt a klíčová slova**

### **Abstrakt:**

Cílem práce je vytvořit obslužnou aplikaci pro měřicí přístroj PM100, od výrobce Thorlabs. Komunikace probíhá přes sériovou linku RS-232. Jako vývojový nástroj byl použit MATLAB verze R2010b. Na monitoru obslužného počítače je pomocí grafického uživatelského prostředí GUIDE vytvořena aplikace. Uživatel tak může pomocí myši, případně klávesnice, ovládat, nastavovat a měnit vlastnosti měřícího přístroje a snadno ukládat hodnoty měření.

### **Klíčová slova:**

**GUI, GUIDE, MATLAB, RS232, PM100, Thorlabs**

### **Abstract:**

Object of my work is creating a service application for measuring unit PM100, from Thorlabs. Communication proceed through serial line RS-232. Creating instrument is MATLAB version R2010b. On monitor tube on service computer is created graphic user interface application in GUIDE. User is able to, using key board or mouse, control all properities and easy save values of measuring.

### **Key words:**

**GUI, GUIDE, MATLAB, RS232, PM100, Thorlabs**

## Obsah

|  |    |
|--|----|
| Zadání.....  | 2  |
| Prohlášení studenta.....                                   | 3  |
| Poděkování.....  | 3  |
| Abstrakt a klíčová slova.....                              | 4  |
| Obsah.....   | 5  |
| Seznam použitých zkratek a symbolů.....                    | 7  |
| Úvod.....  | 8  |
| 1 RS232.....   | 8  |
| 1.01 Obecný popis rozhraní RS-232.....                     | 8  |
| 1.02 Normy RS-232.....                                     | 9  |
| 1.03 Nedostatky standardu.....                             | 9  |
| 1.04 Dnešní využití v osobních počítačích.....             | 10 |
| 1.05 Podrobnější popis standardu a napěťové stupně.....    | 10 |
| 1.06 Konektory a označení pinů.....                        | 11 |
| 1.07 Řízení toku piny RTS/CTS.....                         | 12 |
| 1.08 Přenosové rychlosti při sériovém přenosu dat.....     | 12 |
| 1.09 Popis asynchronní komunikace.....                     | 13 |
| 1.10 Nastavení portu v operačních systémech Windows.....   | 13 |
| 1.11 Nastavení portu v operačních systémech Linux.....     | 14 |
| 1.12 Nastavení RS-232 pro komunikaci s PM-100.....         | 15 |
| 2 PM100.....   | 16 |
| 2.01 Technická data PM100.....                             | 16 |
| 2.02 Formát datových částí vysílaných k měřiči.....        | 18 |
| 2.03 Formát datových částí vysílaných z měřiče.....        | 18 |
| 2.04 IEEE emulační příkazy.....                            | 19 |
| 2.05 Příkazy ze standardu IEEE488.2.....                   | 19 |
| 2.06 Příkazy pro zařízení PM100.....                       | 22 |
| 2.07 Použitý senzor - S120B.....                           | 24 |
| 2.08 Funkce serial v Matlabu.....                          | 25 |
| 3 GUI.....   | 28 |
| 3.01 Tvorba GUI.....                                       | 28 |
| 3.02 Nedostatky aplikace Thorlabs/výhody mé aplikace ..... | 29 |

|   |    |
|---|----|
| 3.03 Grafické rozhraní aplikace pro PM100 v MATLAB..... | 31 |
| 4 Závěr.....  | 40 |
| 5 Použitá literatura.....                               | 41 |
| Příloha.....  | 42 |

## Seznam použitých zkratk a symbolů

| Zkratka        | Význam   |
|----------------|--|
| <b>ASCII</b>   | American Standard Code for Information Interchange (americký standardní kód pro výměnu informací) - v podstatě jde o kódovou tabulku definující znaky anglické abecedy a další znaky.      |
| <b>USB</b>     | Universal Serial Bus - je sériová sběrnice, mimo jiné, pro připojení periférií k počítači  |
| <b>RS-232</b>  | Standard sériové linky z roku 1969.  |
| <b>UART</b>    | Universal Asynchronous Receiver and Transmitter (univerzální asynchroní přijímač a vysílač) - je to zařízení pro sériovou komunikaci, je obsažena i v linkách RS232                        |
| <b>ISO/OSI</b> | Model se používá jako názorný příklad řešení komunikace v sítích. Je to souhrn norem.  |
| <b>DTE</b>     | Data Termina Equipment - označení klientského zařízení v telekomunikacích.   |
| <b>DCE</b>     | Data Communication Equipment - označení řídicího (serverového) zařízení v telekomunikacích   |
| <b>IEEE</b>    | Institute of Electrical and Electronics Engineers (institut pro elektrotechnické a elektronické inženýrství) je mezinárodní organizací usilující o vzestup elektrotechnických technologií. |
| <b>COM</b>     | Hardwarové rozhraní sériového portu.   |
| <b>Baud</b>    | Jednotka modulační rychlosti.  |
| <b>GUI</b>     | Graphical User interface (grafické uživatelské rozhraní) – je to rozhraní umožňující pomocí grafických prvků ovládat počítač.  |
| <b>MATLAB</b>  | Programové prostředí a skriptovací jazyk pro vědeckotechnické účely.   |
| <b>GUIDE</b>   | Grafické uživatelské prostředí v MATLAB, která poskytuje sadu nástrojů pro vytváření GUI.  |

## Úvod

Cílem práce je vytvořit obslužnou aplikaci pro měřicí přístroj PM100, od výrobce Thorlabs. Komunikace probíhá přes sériovou linku RS-232. Jako vývojový nástroj byl použit MATLAB verze R2010b. Komunikace s obslužným PC probíhala přes školní VPN přes program VNC viewer. Doma jsem používal MATLAB verze R2010a, kde jsem programoval části, které nepotřebují nutně přístroj PM100. V první části jsem se zabýval sériovým portem. V druhé jsem se zabýval instrukční sadou PM100. Zprovoznění komunikace bylo stěžejní. Proto je poměrně dlouze zpracované. Třetí fáze práce je tvorba GUI, které by využívalo komunikace s PC. Podrobně jsem popsal všechny segmenty GUI a jejich funkce

## 1 RS-232

### 1.01 Obecný popis rozhraní RS-232

Sériový port se používá ke komunikaci osobních počítačů a další elektroniky. Standard RS-232 definuje elektrické charakteristiky a časování signálu, význam signálu, fyzickou velikost a rozmístění pinů.

RS-232 umožňuje takzvanou sériovou komunikaci dvou zařízení, kdy jednotlivé bity jsou posílány sériově, tzn. za sebou, podobně jako u rozhraní USB nebo technologie ethernet. V současné době se upouští od užívání v osobních počítačích a je nahrazováno univerzálnějším a výkonnějším USB rozhraním. Avšak v průmyslu je tento standard, především jeho modifikace RS-232 a RS-485, velmi rozšířený a pro jeho rysy tomu bude nejspíš i nadále. Na rozdíl od rozhraní USB, RS-232 pouze definuje, jak se má přenést určitá sekvence bitů a nezabývá se vyššími vrstvami komunikace. V referenčním modelu ISO/OSI tak představuje fyzickou vrstvu. Na počítači bývá linka vyvedena konektorem D\_Sub typu DE-9 M a zařízení se připojuje pomocí DE-9 F.



Obrázek č. 1: Konektor RS232

Pro připojení zařízení používajících RS-232 k současným počítačům se buď používají rozšiřující desky, nebo převodníky USB/RS-232. Ty ale mají proti "skutečné" lince RS-232 výrazně horší odezvu, což v některých aplikacích může způsobovat problémy, až nefunkčnost. V mém zapojení budu používat přímo RS-232 bez převodníku.

*V kapitole 1.01 bylo čerpáno z:[6],[7]*



## 1.02 Normy RS-232

Elektronics Industries Association (EIA) standard RS-232-C, uvedený v roce 1969, definuje:

- Elektrické signální charakteristiky jako napěťové úrovně, přenosovou rychlost, překlápění signálů v čase, napěťové hodnoty mimo napěťový level, chování při zkratu a maximální kapacitu mezi vodiči.
- Mechanické charakteristiky rozhraní jako vyndavatelné konektory a definování pinů.
- Funkce všech obvodů v konektoru.
- Standardy vložených obvodů pro vybrané komunikační aplikace.

Standard nedefinuje prvky jako:

- Převod na znaky, například ASCII.
- Rámcování znaků v datový tok (rychlost přenosu, start/stop bity a ani paritní bity).
- Protokoly pro nalezení chyb nebo algoritmy pro kompresi dat.
- Rychlost přenosu v bitech, ačkoli standard říká že rychlost nesmí být vyšší než 20 000 bitů za sekundu. Některé zařízení podporují rychlost 115 200 bitů za sekundu a i více.
- Napájení pro externí zařízení.

Přesný formát přenosu a přenosovou rychlost řídí hardware seriového portu, obvykle jedním vestavěným obvodem UART. Ten mění paralelní data na sériové asynchronní, vytváří přesné napěťové úrovně, a zaručuje rychlost obrácení signálu z (1 na 0 nebo 0 na 1). Zkratové chování jsou obvykle řízeny řadičem, který mění UART logické úrovně na RS-232 logické úrovně a přijmač, který má opačnou funkci.

V kapitole 1.02 bylo čerpáno z:[6]

## 1.03 Nedostatky standardu

Aplikace používající rozhraní RS-232 se rozšířily mnohem dál, než byl jejich původní účel propojit terminál s modemem. Proto je třeba popsat nedostatky a limity rozhraní:

- Velké napěťové překmity a požadavky na pozitivní a negativní napájení komplikují napájení zařízení. Požadavky na napěťové překmity také omezují nejvyšší přenosovou rychlost, kterou je rozhraní schopné komunikovat.
- Jednobitové signalizace ukazují na běžné omezení jako odolnost proti rušení a vzdálenost přenosu.
- Propojení více než dvou zařízení není definované. Navržené spojení více zařízení mělo problémy s rychlostí a kompatibilitou.
- Asymetrická definice spojení dvou zařízení. Vývojář musí určit jejich role, to znamená určit, který konec bude DTE a který DCE.
- Potvrzovací řídící řády jsou určeny pro nastavení a konec komunikace, zejména potvrzovací řády nejsou spolehlivé v mnoha zařízeních.

- Není určena síla vysílače k zařízení. Odběry proudu jsou možné z DTR a RTS pinů, ale to je možné jen u malých odběrů, jako například u myši.
- Doporučený 25-pinový konektor je příliš velký a nepraktický.

*V kapitole 1.03 bylo čerpáno z:[6],[7]*

## 1.04 Dnešní využití v osobních počítačích

Microsoft kritizoval RS-232 v knize PC 97 Hardware Design Guide. V dnešní době se RS-232 v osobních počítačích již téměř nepoužívá a je nahrazeno USB. V porovnání s USB je pomalejší, používá nižší napětí ke komunikaci a konektory se hůře zastrkují. Oba standardy mají softwarovou podporu ve všech rozšířenějších operačních systémech. USB je komplexnější, obsahuje protokol pro přenos dat, ale nedovoluje uživateli přímou analogovou komunikaci. RS-232 pouze definuje napětí signálů a funkci fyzického rozhraní – pinů. Sériový port se používá pro přímé řízení různých hardwarových zařízení, jako jsou relátka, žárovky atd. Během ovládání lze přímo a snadněji komunikovat přes software. Takové úkony nejde provést přes USB, které vyžadují určitou formu přenosu a následně dekodují data. Nelze tam tedy nastavit na delší dobu hodnotu na některém pinu. USB je dnes místo RS-232 na klávesnicích a myších, avšak jsou konvertory, které změní RS-232 na USB. Ty však nemusí spolupracovat s každým softwarem a na všech počítačích, navíc redukuje šířku pásma napětí a mají určité časové zpoždění. Některé počítače používají sériový port jako nepřerušovaný napájecí zdroj nízkého výkonu.

*V kapitole 1.04 bylo čerpáno z:[6],[7]*

## 1.05 Podrobnější popis standardu a napěťové stupně

Data jsou posílány sériově, jak synchronně tak asynchronně, standard podporuje oba způsoby přenosu. Dále definuje přídavné datové obvody, počty řídicích obvodů a obvody řešící spojení DTE a DCE. Všechna data komunikují pouze v jednom směru. Obvody na příjem a vysílání dat jsou oddělené, rozhraní pracuje v režimu full duplex a podporuje datový posun v obou směrech. Standard nedefinuje rámcování v datovém proudu ani překlad znaků.

Standard definuje napěťové úrovně logické jedničky a logické nuly pro přenos dat a napěťové úrovně pro řídicí linky. Schválené napětí je od  $\pm 3$  až  $\pm 15$  voltů. Rozmezí od  $\pm 3$  voltů k nule je neplatné. Maximální napětí naprázdno je 25 voltů, signální úrovně jsou  $\pm 5$  V,  $\pm 10$  V,  $\pm 12$  V a  $\pm 15$  V. Dále je ošetřeno napájení při zkratu a při přivedení napětí vyššího než  $\pm 25$  voltů. RS-232 by toto napětí měla vydržet. Rychlost změn mezi napěťovými úrovněmi je rovněž ošetřena.

Linky pro přenos dat (TxD, RxD a jejich sekundární ekvivalenty) mají logickou jedničku definovanou na minusovém napětí. Logické nuly jsou pak na plusovém napětí. Řídicí signály jsou logicky převrácené. Když je jeden ze signálů aktivní, tak napětí na řídicích linkách je mezi  $+3$  a  $+15$  volty, v opačném případě, kdy je neaktivní, tak je napětí mezi  $-3$  a  $-15$  volty. Příklady řídicích linek jsou: request to send (RTS), clear to send (CTS), data terminal ready (DTR) a data set ready (DSR).

Protože jsou napěťové úrovně běžně vyšší, než mají být, tak RS-232 obsahuje integrované obvody, které toto napětí sniží na úroveň logických úrovní. To chrání další integrované obvody od přepětí, které by je mohlo poškodit, a dále pak poskytuje dostatečný proud pro další práci se signálem.

Napětí na obou koncích je odvozené od nulového vodiče, který je připojen k zemi. Problémy však nastanou, když napětí mezi oběma nulovými vodiči není nula. To může vést k nebezpečnému proudu z jednoho nulového vodiče do druhého. To nás omezuje k používání relativně krátkých kabelů. Napětí mezi nulovými vodiči může být způsobeno i různými napájecími zdroji obou zařízení na koncích kabelu. Tento rozdíl napětí má za následek snížení prostoru pro rozpětí signálu, což je nevýhodné. USB, RS-422 a RS-485 umějí tolerovat větší napětíové rozdíly mezi nulovými vodiči, neboť používají diferenciální signalizaci.

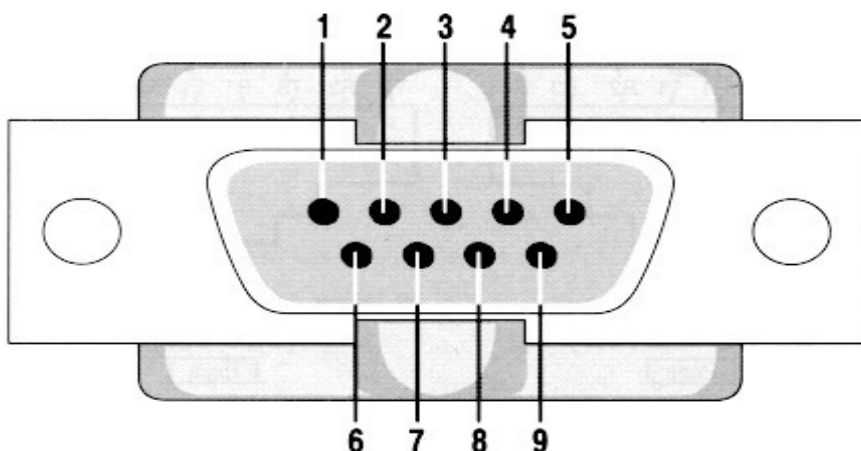
Nevyužité linky připojené k zemi nesmí mít nedefinovaný logický stav. To znamená, že musí být trvale nastaven řídicí signál na logickou jedničku nebo nulu.

*V kapitole 1.05 bylo čerpáno z:[6],[7],[8]*

## 1.06 Konektory a označení pinů

Zařízení jsou rozdělena na DTE a DCE, kde DTE má přirozené označení signálů a DCE má signály opačně, z pohledu DTE-klienta. To definuje, které linky budou přijímat a které vysílat signál. To však neplatí u 25 pinového konektoru. U něj je to sice doporučeno, ale není to nařízeno.

Terminály a počítače mají samčí DTE konektory, modemy mají samičí DCE konektory. Jiné zařízení mohou mít jakoukoli kombinaci pohlaví konektorů a definici pinů. DTE je, zjednodušeně řečeno, koncový prostředek, který převádí informace od uživatele na signály určené k přenosu a nebo převádí přijaté zprávy zpět na digitální informaci. Z pohledu dělení dle iniciativy je DTE klientem, obsluhovaný DCE. DCE je v tomto pohledu něco jako server. Na tomto obrázku jsou jména jednotlivých pinů.



Obrázek č. 2 : Konektor s číselným označením pinů

| Pin | Označení                  | Pin | Označení              |
|-----|---------------------------|-----|-----------------------|
| 1   | Data Carrier Detect (DCD) | 6   | Data Set Ready (DSR)  |
| 2   | Receiver Data (RD)        | 7   | Request To Send (RTS) |
| 3   | Transmitted Data (TD)     | 8   | Clear To Send (CTS)   |
| 4   | Data Terminal Ready (DTR) | 9   | Ring Indicator (RI)   |
| 5   | Signal Ground (GND)       |     |                       |

Tabulka č. 1 : Označení pinů

*V kapitole 1.06 bylo čerpáno z:[6],[8]*

## 1.07 Řízení toku piny RTS/CTS

V dřívějších verzích bylo použití RTS a CTS asymetrické. DTE vysílá sérii bitů, pomocí RTS požadavku k přenosu na DCE a DCE odpovídá a dává svolení ke komunikaci pomocí CTS. To umožní poloduplexním modemům vypnout jejich vysílače, když nejsou potřeba a nařizuje vyslat synchronizační bity, kdykoliv se vysílače znovu aktivují. Tenhle postup se používá i dnes v měničích z RS-232 na RS-485, kde RS-232 signál RTS je použit k dotazu k převzetí řízení toku RS-485. Tento postup se dnes v RS-232 standardu už nepoužívá, dnes už DTE nemůže nijak projevit, že není schopno přijmout přenášená data.

Vyrábějí se i nestandardní symetrické alternativy. CTS indikuje povolení z DCE, do DTE posílá data DCE. RTS indikuje povolení posílat data z DCE do DTE. Tento způsob se používá u standardu RS-232-E.

*V kapitole 1.07 bylo čerpáno z:[7]*

## 1.08 Přenosové rychlosti při sériovém přenosu dat

Běžné přenosové rychlosti jsou odvozeny od násobku 300 bitů za sekundu, ale některé rychlosti starších zařízení, například dálnopisů, měly i nižší rychlosti (například 50, 75 či 150 bitů za sekundu). V následující tabulce jsou uvedeny přenosové rychlosti při nastavení přijmače i vysílače na osm bitů, jednoho start a jednoho stop bitu s délkou odpovídající běžnému bitu. Toto nastavení se běžně označuje jako 8N1. Kvůli přítomnosti start a stop bitu je v podstatě 20 % celkové rychlosti ztraceno. V praxi je však toto zpoždění přijatelné, zařízení se nemusí synchronizovat synchronizačním signálem.

| Přenosová rychlost | Doba přenosu jednoho bitu | Doba přenosu jednoho bajtu |
|--------------------|---------------------------|----------------------------|
| 1200               | 833 $\mu$ s               | 8,33 ms                    |
| 2400               | 416 $\mu$ s               | 4,16 ms                    |
| 4800               | 208 $\mu$ s               | 2,08 ms                    |
| 9600               | 104 $\mu$ s               | 1,04 ms                    |
| 19200              | 52 $\mu$ s                | 520 $\mu$ s                |
| 38400              | 26 $\mu$ s                | 260 $\mu$ s                |
| 115200             | 8,6 $\mu$ s               | 86 $\mu$ s                 |

Tabulka č. 2: Přenosové rychlosti

Na přijmač a vysílač jsou kladeny poměrně přísné požadavky na dodržení přenosové doby jednotlivých bitů. Odchylka rychlosti pro jeden bit nesmí být větší než 2 %, neboť se tato odchylka sčítá. A pro osm datových, jeden start a jeden stop bit odchylka činí 20 %. Dalších 6 % nepřesnosti je způsobeno detekcí počáteční hrany start bitu (to platí pro vzorkování linky šestnáctinásobnou rychlostí, protože 100/16 dává oněch 6 %, neboť mezi dvěma vzorky nelze hranu detekovat). Celková chyba při chybě rychlosti 2 % na konci znamená 26 %. Navíc musí být tři prostřední vzorky v každém bitu stabilní, proto musíme připočíst "oblast jistoty" 18,75 %. Při kolísání

přenosové rychlosti o hodnotu větší než 2 % riskujeme, že tři prostřední vzorky, na jejichž základě se určuje hodnota bitu, se již v posledních bitech přenosu budou nacházet na jeho dolní či horní hranici a né v jeho polovině. To vede k chybám při přenosu dat.

*V kapitole 1.08 bylo čerpáno z:[7],[8]*

## 1.09 Popis asynchronní komunikace

Ikdyž komunikující zařízení znají rychlost, jakou se data přenášejí, musí přijmač začít přijímat ve správný okamžik. Jinými slovy, musí proběhnout synchronizace. V případě synchronní komunikace existuje souběžně s datovým vodičem i vodič synchronizační, na kterém vysílač oznamuje přijmači, že začal vysílat. Naopak u asynchronní komunikace synchronní vodič není, vysílač pouze vyšle předem definovaná data přijmači po datovém vodiči. Po jejichž přijetí se přijmač zasynchronizuje.

V případě RS-232 každé sekvenci datových bitů předchází jeden start bit, kterým se logická hodnota na lince přepne (původně byla v klidovém stavu) do opačného stavu. Potom následují samotná data a po nich paritní bit, podle kterého poznáme, že přenos mohl být v pořádku (v případě 0 nebo sudého počtu chybných bitů) a nebo, že přenos neproběhl v pořádku. Po paritním bitu následuje jeden nebo několik stop bitů, během kterých se linka dostane do klidového stavu. Je tak možné použít méně vodičů na synchronní přenos dat na úkor určitého omezení rychlosti.

*V kapitole 1.09 bylo čerpáno z:[6]*

## 1.10 Nastavení portu v operačních systémech Windows

Většina Intelovských počítačů má alespoň jeden nebo dva sériové porty (Com1 a Com2) ve standardu RS-232C. Pod DOS byl přístup k portům přímý, počínaje Windows 95 byl přímý hardwarový přístup k nastavení portu zrušen. Ve Windows 95 k němu lze přistupovat přes určité nástroje. Pomocný soubor ve Visual Basic nám může pomoci pochopit nastavení driveru Comm.drv. Následující výpis vypisuje příklad nastavení:

|  |                                  |
|--|----------------------------------|
| MSComm1.CommPort = 2                           | ' Nastavení čísla portu          |
| MSComm1.Settings = "56000,N,8,1"               | ' Nastavení UART parametrů       |
|  | '(rychlost v baudech, N jako bez |
|  | 'parity, 1 stop bit)             |
| MSComm1.PortOpen = True                        | ' Zda je povolena možnost        |
| zablokovat port MsComm1.Output = "Text string" | ' Posílaná data                  |
| Buffer\$ = Buffer\$ & MSComm1.Input            | ' Přijímaná data                 |

Na tomto useku driveru vidíme hlavní nastavení portu.

V operačních systémech Windows 9x, NT, 2000 a XP jsou všechny vstupně/výstupní porty reprezentovány jako soubor, takže práce s nimi jsou obstarávány funkcemi jako „vytvoř soubor“, „čti soubor“ atd. (CreateFile, CloseHandle, ReadFile, ReadFileEx, WriteFile a WriteFileEx). Tyto funkce otevírají a zavírají spojení, poskytují hlavní popis rozhraní, nastavují konfiguraci, čtou a nebo zapisují na port a zachycují informace o hlavních událostech na portu.

*V kapitole 1.10 bylo čerpáno z:[8]*

## 1.11 Nastavení portu v operačních systémech Linux

V unixových operačních systémech jsou sériové porty reprezentovány soubory: /dev/ttyS0 jako COM1, /dev/ttyS1 jako COM2. Dají se otevřít jako soubor a lze do nich zapisovat a číst z nich jako ze souborů. Není třeba žádných dalších knihoven. Lze v nich hned měnit rychlost a ostatní nastavení. Zde je příklad výpisu konfigurace souboru /dev/ttyS0 reprezentující sériový port.

```
setserial -ag /dev/ttyS0
/dev/ttyS0, Line 0, UART: 16550A, Port: 0x03f8, IRQ: 4
  Baud_base: 115200, close_delay: 50, divisor: 0
  closing_wait: 3000, closing_wait2: infinte
  Flags: spd_normal skip_test session_lockout
```

Práce s portem na linuxu má i výhody, například ve větším množství přenosových rychlostí. Linux nabízí jak velmi nízké rychlosti, tak velmi vysoké, Windows nenabízí.

| Linux | Windows | Linux   | Windows |
|-------|---------|---------|---------|
| 50    | není    | 19200   | 19200   |
| 75    | není    | 38400   | 38400   |
| 110   | 110     | 57600   | 57600   |
| 134   | není    | 115200  | 115200  |
| 15    | není    | 230400  | 128000  |
| 200   | není    | 460800  | 256000  |
| 300   | 300     | 500000  | není    |
| 600   | 600     | 576000  | není    |
| 1200  | 1200    | 576000  | není    |
| 1800  | není    | 921600  | není    |
| 2400  | 2400    | 1000000 | Není    |
| 4800  | 4800    | 19200   | 19200   |
| 9600  | 9600    |         |         |

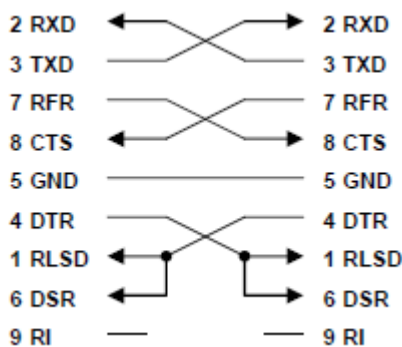
Tabulka č. 3.: Přenosové rychlosti v OS Linux a Windows

*V kapitole 1.11 bylo čerpáno z:[8]*

## 1.12 Nastavení RS-232 pro komunikaci s PM-100

Zařízení má rozhraní které odpovídá IEEE1174 standardu. Umožňuje emulovat běžné IEEE488 rozhraní na sériové lince a používá SCPI styl příkazů. Zařízení je považováno za DTE. K propojení s počítačem se používá 9-pólový null-modem kabel. Jedna strana se zapojí do DB9 socketu na vrchu zařízení PM100 a druhá do volného COM portu na PC.

Sériový způsob nulového zapojení je uveden v následující tabulce; čísla indikují piny DB9 konektoru.



Obrázek č. 3: Propojení pinů mezi dvěma konci RS-232

### Nastavení PC pro komunikaci:

|                             |   |
|-----------------------------|---|
| Nastavení linky:            | 2400 .. 115200 – 8N1 – RFR/CTS<br>(8 datových bitů, 1 stop bit, bez parity) |
| Programové ukončení zprávy: | <LF><br>(Line Feed, 0Ahex)  |
| Poslední znak odpovědi:     | <CR><LF><br>(Carriage Return, 0Dhex; Line Feed, 0Ahex)                      |

*V kapitole 1.12 bylo čerpáno z:[3]*

## 2 PM100

### 2.01 Technická data PM100

#### Hlavní specifikace:

|                     |   |
|---------------------|---|
| Displej:            | 240 x 160 pixelů LCD s EL podsvícením                                       |
| Formát displeje:    | číslicový, analogový – bodový, sloupcový diagram, statistiky a screeny menu |
| Komunikace s PC:    | RS-232  |
| Rychlost v Baudech: | 2400 – 115200 bps   |



Obrázek č. 4: Měřicí přístroj PM100

#### Měřič jednotky:

|                                      |                                 |
|--------------------------------------|---------------------------------|
| Proudový rozsah foto-diody:          | 0.1nA – 4mA                     |
| Napěťový rozsah teplotního senzoru : | 1μV – 10mV                      |
| Přesnost:                            | ±1 % (konzole)                  |
| Přesnost:                            | 16 až 24 bitů (ΣΔA/D konvertor) |
| Vzorkovací frekvence:                | 6 Hz                            |

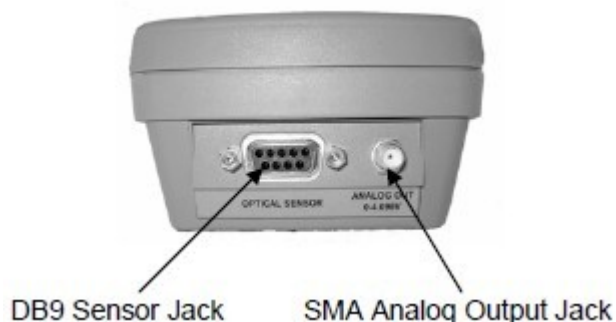


### Analogový výstup:

|                       |   |
|-----------------------|---|
| Napěťový rozsah:      | 0 – 4,095 V                                   |
| Rozlišení:            | 12 bitů (1mV)                                 |
| Vzorkovací frekvence: | 6 Hz  |
| Výkonový rozsah:      | 0,001V/mW – 1000V/mW, Programově nastavitelný |
| Konektor:             | SMA   |

### Napájení:

|                |                                      |
|----------------|--------------------------------------|
| Baterie:       | NiMH, 1500mAh, 3,6V                  |
| Výdrž baterie: | > 12 hodin (při vypnutém podsvícení) |
| Nabíječka:     | ano                                  |
| Napájení:      | Vstup: 85 – 264 VAC, 50 – 60 Hz      |
| Výstup:        | 9 VDCV @ 1.1A                        |



Obrázek č. 5: Měřicí přístroj PM100 pohled shora



Obrázek č. 6: Měřicí přístroj PM100 pohled ze spodu

*V kapitole 2.01 bylo čerpáno z:[3]*

## 2.02 Formát datových částí vysílaných k měřiči

Řídící PC posílá datové položky na zařízení, definované IEEE488.2-1992 a tím ho ovládá. Zařízení přijímá následující položky:

### <Vlastnost>

Prvek je používán k vyjádření informace o parametru. Tento element nerozlišuje malá a velká písmena. Například: ON , Off , ENABLE , Disable

### <Dekadické číslo>

Prvek se používá k vyjádření číselné hodnoty na zařízení. Znaky mimo tento element nejsou přípustné. Například: 5 , -23.5 , 123E-3 , +0.123

### <Nedekadické číslo>

Prvek se používá k vyjádření číselné hodnoty integerem. Znaky mimo tento element nejsou přípustné. Například: #H1E , #036 , #B11110 (hexadecimální, osmičkové, binární - vyjádření 30)

### <Řetězec>

Element obsahuje všechny prvky ASCII tabulky. Například: Hello world , "Hello word"

*V kapitole 2.02 bylo čerpáno z:[3]*

## 2.03 Formát datových částí vysílaných z měřiče

Zařízení vysílá výsledky měření, nastavení nebo informace o statusech. Data bývají definované v IEEE488.2-1992. Zařízení vysílá následující typy položek:

### <Vlastnost>

Prvek je používán k vyjádření informace o parametru. Tento element nerozlišuje malá a velká písmena. Například: ON , Off , ENABLE , Disable

### <Celé číslo>

Používá se k přenosu číselné hodnoty. Například: 5, -23

### <Desetinné číslo>

Používá se k přenosu přesného číselného údaje. Například: 5.0, -23.1

### <Exponenciální číslo>

Reprezentuje číslo - jeho základ s polečně s exponentem. Například : 5.0E+03 , -23.1E-06

### <Šestnáctkové číslo>

Je určeno k přenosu šestnáctkového čísla. Například: #H1E

### <Řetězec>

Element obsahuje všechny prvky ASCII tabulky. Například: Ahoj word , "Hello word"

*V kapitole 2.03 bylo čerpáno z:[3]*

## 2.04 IEEE emulační příkazy

Následující emulační příkazy jsou zařízením PM100 podporovány, ostatní, z IEEE1174, nejsou.

| Emulační příkaz | Směr         | Popis   |
|-----------------|--------------|---|
| &SRQ<CR><LF>    | PM100 --> PC | Informuje řídicí zařízení (PC), že vyžaduje obsluhu.                |
| &POL            | PC --> PM100 | Nastaví se Status byte.   |
| &ddd<CR><LF>    | PM100 --> PC | Odpovídá '&POL'. 'ddd' je 3 místná číselná hodnota ze Status bajtu. |

Tabulka č. 4: Emulační příkazy

Funkce SQR má následující průběh:

1. Zařízení vyšle '&SRQ<CR><LF>' na řídicí jednotku a tím signalizuje, že chce být obslouženo.
2. Řídicí PC odpoví '&POL' jako dotaz do Status bajtu.
3. Zařízení odpoví '&ddd<CR><LF>', kde 'ddd' je třímístná digitální desítková hodnota Status bajtu.

*V kapitole 2.04 bylo čerpáno z:[3]*

## 2.05 Příkazy ze standardu IEEE488.2

Zařízení podporuje komunikační protokol MEC verze IEEE488.2. Neposílá data dokud o ně není požádáno (neplatí pro emulační kódy). Příkazů se dá psát více najednou a musí se oddělit středníkem. Když je příkaz zakončen „?“ znamená to, že zařízení bude odpovídat nějaké data. Příkaz bez otazníku nemá odpověď.

| Příkaz | Stručný význam                             |
|--------|--|
| *IDN?  | Identifikace zařízení.                     |
| *TST?  | Autotest, jestli je zařízení v pořádku.    |
| *OPC   | Nastavení OPC bity v SESR registru.        |
| *OPC?  | Zjištění hodnoty OPC bitu v SESR registru. |
| *WAI   | Chvilková nečinnost.                       |
| *RST   | Reset zařízení.                            |
| *SRE   | Natavení registru SRE.                     |
| *SRE?  | Stav SRE registru.                         |
| *STB?  | Stav Status bajtu.                         |
| *ESE   | Nastavení ESE registru.                    |
| *ESE?  | Stav ESE registru.                         |
| *ESR?  | Stav ESR registru.                         |
| *CLS   | Vynulování registrů.                       |

Tabulka č. 5: Příkazy ze standardu IEEE

### Popis jednotlivých příkazů

|                  |   |
|------------------|---|
| Příkaz:          | <b>*IDN?</b>  |
| Formát odpovědi: | Řetězec   |
| Popis:           | Identifikace zařízení jako: výrobce, model, sériové číslo atd.                          |
| Příklad:         | 'Thorlabs, PM100, M00219695, Rev. 2.00 (Mar 14 2006)'                                   |
|                  |   |
| Příkaz:          | <b>*TST?</b>  |
| Formát odpovědi: | Celé číslo  |
| Popis:           | Autotest zařízení, když proběhlo všechno v pořádku, odpoví '0'.                         |
| Příklad:         | '0'   |
|                  |   |
| Příkaz:          | <b>*OPC</b>   |
| Popis:           | Nastavení na '1' bit OPC v ESR registru.  |
|                  |   |
| Příkaz:          | <b>*OPC?</b>  |
| Formát odpovědi: | Celé číslo  |
| Popis:           | Zjištění stavu bitu OPC v ESR registru.   |
| Příklad:         | '1'   |
|                  |   |
| Příkaz:          | <b>*WAI</b>   |
| Popis:           | Příkaz nemá pro toto zařízení smysl, přikáže zařízení aby po určitou dobu nic nedělalo. |
|                  |   |
| Příkaz:          | <b>*RST</b>   |
| Popis:           | Zařízení se resetuje do výchozí polohy uložení na adrese #1.                            |
|                  |   |
| Příkaz:          | <b>*SRE</b> <Celé číslo>  |
| Popis:           | Nastavení SRE registru.   |
|                  |   |
| Příkaz:          | <b>*SRE?</b>  |
| Formát odpovědi: | Celé číslo  |
| Popis:           | Hodnota SRE registru.   |
| Příklad:         | '0'   |
|                  |   |
| Příkaz:          | <b>*STB?</b>  |
| Formát odpovědi: | Celé číslo  |
| Popis:           | Dotaz na hodnotu Status bajtu.  |
| Příklad:         | '4'   |
|                  |   |
| Příkaz:          | <b>*ESE</b> <Dekadické číslo>   |
| Popis:           | Nastavení ESE registru.   |

Příkaz: **\*ESE?**  
Formát odpovědi: Celé číslo  
Popis: Stav ESE registru.  
Příklad: '0'

Příkaz: **\*ESR?**  
Formát odpovědi: Celé číslo  
Popis: Stav ESR registru.  
Příklad: '0'

Příkaz: **\*CLS**  
Popis: Příkaz vynuluje následující registry: ESR, DEE, DEEE, DSE a DSEE

V příkladu jsem vždy vypsál to, co mi na dotaz odpověděl přístroj PM100, v jednom z nastavení. K přístroji byl vždy připojen diodový senzor S120B.

*V kapitole 2.05 bylo čerpáno z:[3]*

## 2.06 Příkazy pro zařízení PM100

| Příkaz                    | Popis  |
|---------------------------|--|
| <b>:POWER?</b>            | Poslední měřená hodnota výkonu ve watech.  |
| <b>:HEAD:INFO?</b>        | Informace na měřicím čidle jako ID, popis, maximální výkon.                                    |
| <b>:WAVELENGTH?</b>       | Vlnová délka světla, na které jsou prováděny měření - v metrech.                               |
| <b>:WAVELENGTH</b>        | Nastavení vlnové délky v metrech na které mají být prováděna měření.                           |
| <b>:WAVELENGTH:RANGE?</b> | Rozsah vlnových délek, na kterých je možno provádět měření.                                    |
| <b>:FILTER?</b>           | Nastavení napájecí frekvence, jež má být odmítána.   |
| <b>:FILTER</b>            | Dotaz na zjištění napájecí frekvence, která je odmítána.                                       |
| <b>:PHOTOCURRENT?</b>     | Poslední měřená hodnota proudu v čidle – platí jen pro foto-diodové senzory.                   |
| <b>:PHOTOVOLTAGE?</b>     | Poslední měřená hodnota napětí na čidle – platí jen pro teplotní senzory.                      |
| <b>:DARKCURRENT?</b>      | Zjištění proudu přiváděného do čidla – platí jen pro foto-diodové senzory.                     |
| <b>:DARKCURRENT</b>       | Nastavená trvalého proudu přiváděného do čidla – platí jen pro foto-diodové senzory.           |
| <b>:DARKVOLTAGE?</b>      | Zjištění přiváděného napětí na čidle bez světla – platí jen pro teplotní senzory.              |
| <b>:DARKVOLTAGE</b>       | Nastavená trvalého proudu přiváděného na čidlo – platí jen pro teplotní senzory.               |
| <b>:ATTENUATION?</b>      | Zjištění korekce zesílení nebo útlumu v decibelech – při přidání přídatného skleněného filtru. |
| <b>:ATTENUATION</b>       | Nastavení zesílení nebo útlumu v decibelech – při přidání přídatného skleněného filtru.        |

Tabulka č. 6: Specifické příkazy PM100

|                  |  |
|------------------|--|
| Příkaz:          | <b>:POWER?</b>   |
| Formát odpovědi: | Exponenciální číslo  |
| Popis:           | Dotaz na zjištění posledního měřeného výkonu ve watech.      |
| Příklad:         | '5.936575e-09'   |
| Příkaz:          | <b>:HEAD:INFO?</b>   |
| Formát odpovědi: | <Řetězec><br><Řetězec><br><Řetězec><br><Exponenciální číslo> |

|                  |  |
|------------------|--|
| Popis:           | Identifikace senzoru. První řetězec udává sériové číslo, druhý název, třetí popis a číslem je vyjádřen maximální výkon. Údaje jsou uloženy v EEPROM v čidle. |
| Příklad:         | '07070423, S120B, D 50mW, 5.000000e-02'  |
| Příkaz:          | <b>:WAVELENGTH?</b>  |
| Formát odpovědi: | Exponenciální číslo  |
| Popis:           | Dotaz na použitou vlnovou délku v metrech. Na této vlnové délce je snímán výkon.   |
| Příklad:         | '4.580000e-07'   |
| Příkaz:          | <b>:WAVELENGTH &lt;Dekadické číslo&gt;</b>   |
| Popis:           | Nastavení vlnové délky v metrech, ve které má být prováděno měření.  |
| Příkaz:          | <b>:WAVELENGTH:RANGE?</b>  |
| Formát odpovědi: | <Exponenciální číslo><br><Exponenciální číslo>   |
| Popis:           | Minimální a maximální vlnová délka, na kterou lze čidlo nastavit, udáváno v metrech.   |
| Příklad:         | '4.000000e-07, 1.100000e-06'   |
| Příkaz:          | <b>:FILTER?</b>  |
| Formát odpovědi: | 50HZ 60HZ  |
| Popis:           | Dotaz, která vlnová délka je filtrována, v Evropě by měla být nastavena na 50 Hz, neboť v síti je 60 Hz.   |
| Příklad:         | '50HZ'   |
| Příkaz:          | <b>:FILTER 50HZ 60HZ</b>   |
| Popis:           | Nastavení frekvence, jenž má být odmítána. 50 nebo 60 Hz.  |
| Příkaz:          | <b>:PHOTOCURRENT?</b>  |
| Formát odpovědi: | Exponenciální číslo  |
| Popis:           | Poslední hodnota změřeného proudu v ampérech na senzoru – jen pro foto-diodové senzory.  |
| Příklad:         | '4.937766e-12'   |
| Příkaz:          | <b>:PHOTOVOLTAGE?</b>  |
| Formát odpovědi: | Exponenciální číslo  |
| Popis:           | Poslední hodnota změřeného napětí ve voltech na senzoru – jen pro teplotní čidla.  |
| Příklad:         | '0.000000e-01'   |
| Příkaz:          | <b>:DARKCURRENT?</b>   |
| Formát odpovědi: | Exponenciální číslo  |

Popis: Zjištění nastaveného proudu, jenž má protékat čidlem i bez světelného zdroje, které proud produkuje, v ampérech. Platí jen pro křemíkové nebo germaniové diody – to není případ mého senzoru.

Příklad: '0.000000e-01'

Příkaz: **:DARKCURRENT** <Dekadické číslo>

Popis: Nastavení proudu do diody v ampérech.

Příkaz: **:DARKVOLTAGE?**

Formát odpovědi: Exponenciální číslo

Popis: Zjištění nastaveného napětí, které má být přiváděno na senzor i bez světelného zdroje – ve voltech. Platí jen pro teplotní čidla.

Příklad: '0.000000e-01'

Příkaz: **:DARKVOLTAGE** <Dekadické číslo>

Popis: Nastavení napětí, které má být přiváděno na senzor – platí jen pro teplotní čidla. Toto napětí ve voltech není zahrnováno do výsledku měření.

Příkaz: **:ATTENUATION?**

Formát odpovědi: Exponenciální číslo

Popis: Dotaz na hodnotu zesílení/zeslabení v decibelech. Toto zesílení/zeslabení může být způsobeno přídavným skleněným filtrem na čidle.

Příklad: '0.000000e-01'

Příkaz: **:ATTENUATION** <Dekadické číslo>

Popis: Nastavení hodnoty zesílení/zeslabení v decibelech.

V příkladu jsem vždy vypsál to, co mi na dotaz odpověděl přístroj PM100 na jednom nastavení. K přístroji byl vždy připojen diodový senzor S120B, tudíž dotazy na nastavení teplotního senzoru nemají smysl, ale jsou zde i přesto uvedeny. Stejně tak čidlo pracuje od nulového proudu a není třeba do něj přivádět žádný konstantní proud a pracuje bez přídavného filtru na zesílení/zeslabení výkonu.

*V kapitole 2.01 bylo čerpáno z:[3]*

## 2.07 Použitý senzor - S120B

S120B je optický, výkonový senzor, navržený pro přímé spojení se zařízeními PM100 a PM30. Měří světlo na hranici viditelnosti až k ultračervenému světlu, tedy 400 až 1100 nanometrů. Měří výkon od 50 nW až po 50 mW. Paměť EEPROM je umístěna u DB9 konektoru. Samotná foto-dioda je vyrobena ze silikonu a dají se na ni monovat různé filtry, optické adaptéry a podobně.



### Specifikace:

|                     |                     |
|---------------------|---------------------|
| Vlnová délka:       | 400-1100nm          |
| Materiál senzoru:   | Silikon             |
| Rozměry senzoru     | 10 x 10mm           |
| Výkon:              | 50nW – 50mW         |
| Rozlišení:          | 0.1nW               |
| Měřicí standard:    | NIST                |
| Odchylka:           | +/- 5%              |
| Operační teplota:   | 5 až 40°C           |
| Skladovací teplota: | -20 až +70°C        |
| Poškození při:      | 50W/cm <sup>2</sup> |



Obrázek č. 7: Optický senzor S120B

S120B je kompatibilní s PM100, až od verze firmware 1.62, což však přístroj, se kterým pracuji, splňuje(verze 2.0).

*V kapitole 2.01 bylo čerpáno z:[2]*

## 2.08 Funkce serial v Matlabu

Komunikaci s RS-232 lze popsat v pěti krocích:

1. Vytvoření objektu reprezentující sériový port - vytváří se přes funkci `s = serial()`; přičemž v závorce je uveden název sériového portu, se kterým se má komunikovat, tzn. COM1 nebo COM2 pro Windows. Funkce pak bude třeba vypadat `s = serial('COM1')`; Konfigurace nastavení sériového portu se provádí, až po vytvoření objektu.
2. Připojení k zařízení – připojení se provádí přes vytvořený objekt sériového portu, v našem případě `s`. Funkce tedy vypadá `fopen(s)`; a tím se otevře spojení. Až po otevření spojení se mohou číst a zapisovat data z portu.
3. Nastavení portu – k nastavení jednotlivých vlastností portu se používá funkce `set`, pro zjištění nastavení funkce `get`. Může se provádět buď před, nebo kdykoli po tvorbě objektu reprezentující sériový port.
4. Čtení a zápis dat – data se zapisují pomocí funkcí `fprint` nebo `fwrite`, číst se dají pomocí funkcí `fgetl`, `fgets`, `fread`, `fscan` nebo `readasync`. Sériový port se chová podle předem nakonfigurovaných nebo výchozích nastavení.
5. Ukončení spojení – ve chvíli, kdy necheme dále využívat sériový port, zavřeme jej pomocí funkce `fclose`, smažeme z paměti pomocí `delete` a smažeme z MATLAB prostředí pomocí příkazu `clear`.

Zjištění vlastností připojení:

Pomocí funkce *serial* si vytvoříme objekt a následně, pomocí *get* zjistíme jeho vlastnosti. Zde je výpis vlastností, které má už funkční komunikace s přístrojem PM100.

```
get(s);
ByteOrder = littleEndian
BytesAvailable = 0
BytesAvailableFcn =
BytesAvailableFcnCount = 48
BytesAvailableFcnMode = terminator
BytesToOutput = 0
ErrorFcn =
InputBufferSize = 512
Name = Serial-COM2
ObjectVisibility = on
OutputBufferSize = 512
OutputEmptyFcn =
RecordDetail = compact
RecordMode = overwrite
RecordName = record.txt
RecordStatus = off
Status = closed
Tag =
Timeout = 10
TimerFcn =
TimerPeriod = 1
TransferStatus = idle
Type = serial
UserData = []
ValuesReceived = 0
ValuesSent = 0
```

SERIAL specific properties:

```
BaudRate = 115200
BreakInterruptFcn =
DataBits = 8
DataTerminalReady = on
FlowControl = none
Parity = none
PinStatus = [1x1 struct]
PinStatusFcn =
Port = COM2
ReadAsyncMode = continuous
RequestToSend = on
StopBits = 1
Terminator = LF
```

Tučně jsem zvýraznil nastavení, které je nutné pro správnou komunikaci s přístrojem. Datových bitů musí být 8, bez parity, jeden stop bit, ukončení zprávy musí být LF a rychlost je 115200 baudů. Port je nastaven správně.

Funkce *get* se dá použít i na konkrétní věc, kterou chceme zjistit. Například:

```
get(s,{'Parity','TransferStatus'})
```

a odpovědí je

```
ans =
```

```
'none' 'idle'
```

Tím jsem zjistili paritu a TransferStatus. Lze použít i jinou syntaxi funkce *get*:

```
s.Parity
```

```
ans =
```

```
none
```

### **Nastavení vlastností připojení:**

K nastavení jednotlivých parametrů se používá funkce *set*. Lze použít jak na jednu vlastnost *set(s,'BaudRate',4800)*; nebo jinou formou *s.BaudRate = 4800*; čímž se nastavuje rychlost v baudech na 4800 baudů za vteřinu. Nebo i více vlastností jedním příkazem: *set(s,'DataBits',7,'Name','Test1-serial')*. Funkce si příliš nepotrpí na přesnou syntaxi, což usnadňuje práci. Přijme například *set(s,'BaudRate',4800)* nebo *set(s,'baudrate',4800)* nebo i *set(s,'BAUD',4800)*. Lze nastavovat i dalším způsobem: *set(s,'Parity') [ {none} | odd | even | mark | space ]* kde je hodnota, co má být nastavena, označená složenými závorkami {}.

### **Tvorba objektu:**

Objekty jsou specificky pojmenovány v každém operačním systému. Nejdřív se definuje název spojení a potom jméno portu. V tomto případě je název objektu „obj“ a název portu port. *obj = serial('port')* : v linuxu *serial('/dev/ttyS0')*; v operačním systému MAC *serial('/dev/tty.KeySerial1')*; v operačním systému solaris *serial('/dev/term/a')*; a ve Windows *serial('com1')*; .

*V kapitole 2.08 bylo čerpáno z:[4],[5]*

## 3 GUI

### 3.01 Tvorba GUI

Grafické uživatelské rozhraní bylo vytvářeno ve vývojovém prostředí MATLAB pomocí grafické uživatelského rozhraní GUIDE. V něm se vytvořily uicontroly (tlačítka, graf, textová pole apod.) a zároveň se vytvořil m soubor s prázdnými funkcemi, které reprezentují callbacky. To jsou funkce, které se vykonají vždy po akci uživatele, tzn. stisk konkrétního tlačítka spustí vždy konkrétní funkci. V samotném m souboru není v kódu zanesena podoba grafické podoby, vytvoří se tedy i druhý soubor - .fig.

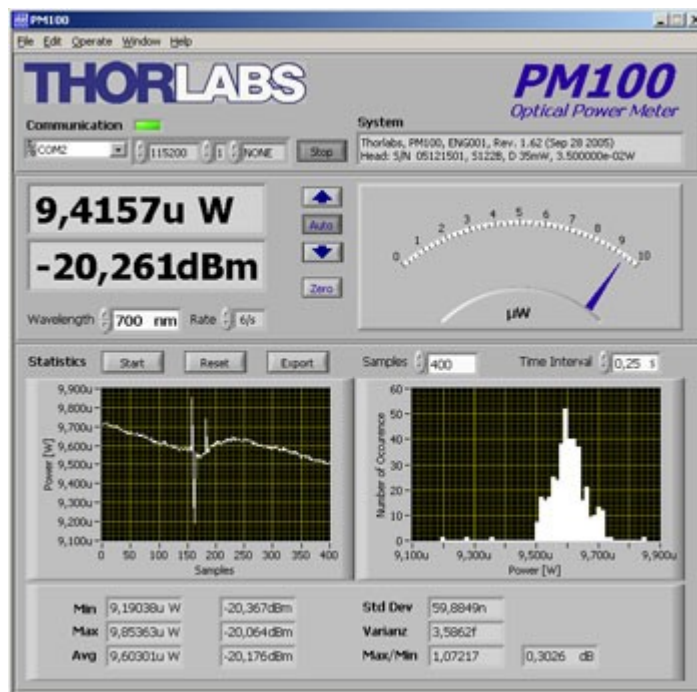
GUIDE automaticky nabízí ve funkcích uživateli tři proměnné. První je hObject, který reprezentuje uicontrols prvky (tlačítka, graf apod.). Druhou proměnnou jsou eventdata, která jsou prázdná a budou snad zaplněny v pozdějších verzích MATLAB. Třetí proměnnou je struktura handle. Do ní by si uživatel měl ukládat všechny proměnné, které mají být známy i v jiných callbackcích. Problémem třetí proměnné je, že při vykonání callbacku se nahrají všechny tři proměnné jako vstupní parametry a nejsou dále aktualizovány, i když se změnily v jiném callbacku. To není problém až do chvíle, kdy se kód nepozdrží v některém z callbacků. Jedná se o reálný čas. Z toho důvodu v inkriminovaných částech nemohu používat struktury handles a musím použít globální proměnné.

*V kapitole 3.01 bylo čerpáno z:[4],[5]*

### 3.02 Nedostatky aplikace Thorlabs/výhody mé aplikace

Při návrhu grafické podoby jsem bral inspiraci z originálního programu od Thorlabs. Avšak musel jsem vylepšit několik věcí. Při celkovém srovnání má moje GUI podstatně méně prvků. Hlavní věc, která mě k tomuto kroku vedla, byla ta, že obslužný počítač nebývá příliš výkonný.

Rozlišení monitoru bylo daleko menší, než rozměr okna originální aplikace. Proto jsem své GUI podstatně zmenšil, zhruba na 1/3 plochy. Z toho důvodu jsem do mé aplikace nedal všechny prvky, které jsou zde:



Obrázek č. 8: Vzhled aplikace pro PM100 od Thorlabs

Dalším nedostatkem byla diagnostika komunikačních problémů. Z tohoto univerzálního hlášení se nedají zjistit žádné informace. Problémem může být nezastrčený kabel, nebo COM užívaný jinou aplikací. Každá z možností vyžaduje jinou akci. Můj program je schopen tyto dvě události rozlišit.



Obrázek č. 9: Hlášení komunikačního erroru

Snad nejdůležitějším rozdílem je formát exportu. Aplikace od Thorlabu dovoluje pouze jeden formát, který je dost specifický a nejde otevřít bez nainstalované aplikace. To podstatně komplikuje následné zpracování dat. Moje aplikace umožňuje dva druhy formátu. První je formát csv, kde je na prvním řádku pojmenování sloupců. Druhý formát je mnohem užitečnější. Jedná se o matici s časem, vlnovou délkou a výkonem. Vše se ukládá do mat-souboru. Z něj lze data do MATLAB snadno načíst a dál zpracovávat.

#### **Ukázka prvního formátu exportu:**

PM100 Optical power meter vytvořeno: 26-Apr-2011 23:26:04

čas,vlnová délka[nm],výkon[W]

23:26:5.171,450,1.491901e-10

23:26:5.671,450,2.037718e-10

23:26:6.296,450,2.650246e-10

23:26:6.812,450,5.433914e-10

23:26:7.281,450,0.000000e-01

23:26:7.765,450,3.408326e-10

23:26:8.203,450,2.547147e-11

23:26:8.718,450,5.585530e-10

23:26:9.203,450,1.946748e-10

23:26:9.703,450,4.675835e-10

23:26:10.171,450,0.000000e-01

23:26:10.671,450,6.743875e-10

23:26:11.156,450,2.389467e-10

23:26:11.812,450,3.475037e-10

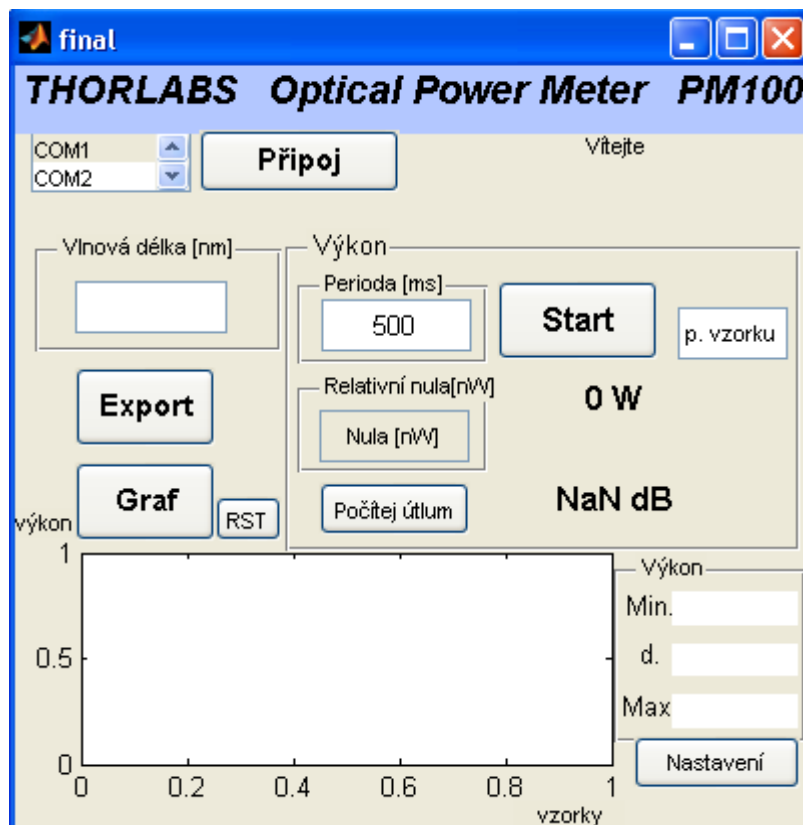
23:26:12.171,450,6.034313e-10

23:26:12.765,450,5.106423e-10

### 3.03 Grafické rozhraní aplikace pro PM100 v MATLAB

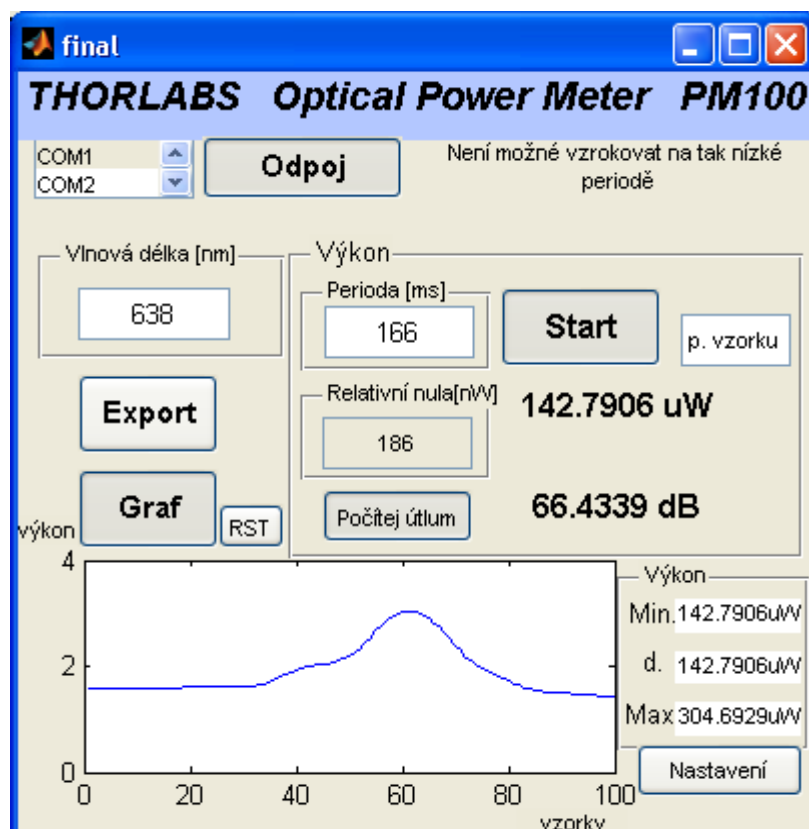
Jak již bylo zmíněno, aplikace byla vytvářena v GUIDE a měla být lepší než originální aplikace od Thorlabs. Toto je vzhled po spuštění kódu m souboru, nebo přímo po spuštění fig souboru.

Cílem bylo, aby uživatel nepotřeboval žádnou nápovědu a aby GUI bylo jednoduché, intuitivní a zároveň umožňovalo co nejvíce nastavení.



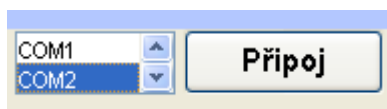
Obrázek č. 10: vzhled GUI po spuštění

Takto vypadá spuštěná aplikace během měření:



Obrázek č. 11: Vzhled GUI během měření

Podrobný popis připojení:



Obrázek č. 12: Vzhled volby COM a tlačítka Připoj

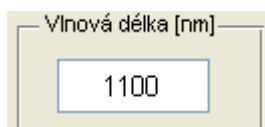
Po volbě COM a stisku tlačítka *připoj* se obslužné PC pokusí spojit se zařízením na druhém konci sériové linky. Nejdřív se MATLAB pokusí vytvořit sériový objekt. PC nemusí mít daný COM vůbec k dispozici. Z důvodu, že není v PC vůbec namontován, nebo sériový port využívá jiná aplikace. Obě varianty se uživatel dozví ve výpisu v GUI, viz obrázek č. 14. Po vytvoření sériového objektu se ihned k zařízení zašle dotaz „IDN?“. Je to dotaz na informace o zařízení. Pokud nepřijde odpověď, zařízení nejspíše není vůbec připojeno. Uživatel je tedy informován ve výpisu. V případě odpovědi je zařízení připojeno. Tlačítko se zamáčkne a nápis se změní na *Odpoj*. Následujícím stiskem se spojení uzavře.





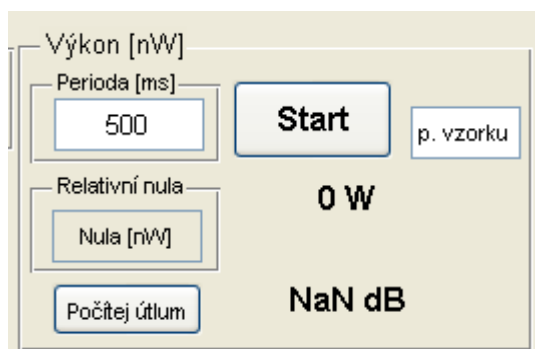
Obrázek č. 13: Vzhled GUI během měření

V této části GUI je uživatel informován o všech důležitých hlášeních. V první fázi je informován o chybách při vytváření spojení. Při úspěšném připojení se vypíší informace o měřicím přístroji a zapojeném senzoru. V dalších fázích se vypisují hlášky, když se uživatel snaží udělat něco nepřípustného, třeba nastavuje vlnovou délku mimo měřicí rozsah senzoru, nebo nastavuje příliš nízkou periodu měření. Vypisují se i některé úspěšné nastavení.



Obrázek č. 14: Vzhled políčka pro nastavení vlnové délky

Editovatelné pole slouží k nastavování vlnové délky v nanometrech. Po připojení přístroje se do něj vloží aktuální délka, na kterou je momentálně měřicí přístroj nastaven. Při přepsání hodnoty je nejprve zkontrolováno, zda je vlnová délka v rozsahu senzoru. V případě, že je v rozsahu, vlnová délka se nastaví. V případě zadání nečíselné hodnoty se vlnová délka nemění. Pokud uživatel zadá délku nižší než je minimální délka senzoru, automaticky se nastaví délka nejbližší hodnotě zadané uživatelem. To je délka minimální. Uživatel je o všech věcech informován v textovém výpisu. Totéž platí o překročení rozsahu na opačné straně.



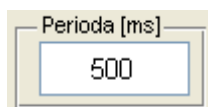
Obrázek č. 15: Vzhled části GUI s měřicími elementy

V této části se spouští a nastavuje periodické měření a počítá útlum.



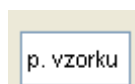
Obrázek č. 16: Tlačítko *Start*

Tlačítko *Start* spouští periodické měření v případě, že je vytvořeno spojení. Při stisku tlačítka se načte perioda měření, která je implicitně 0,5 sekundy a načte se i počet vzorků. Ten je implicitně nastaven na nekonečno. Bylo snahou vytvořit program co nejjednodušší, snadno ovladatelný a intuitivní. Proto stačí dvě tlačítka ke spuštění měření. Jedno je nevyhnutelné, tím je volba COM a tlačítko *Připoj*. Tlačítko *Start* by se mohlo spustit hned po připojení, ale rozhodl jsem se to neudělat. V případě, že se připojením testuje jen přítomnost přístroje, už by nikdo měření nezastavil a zbytečně by byl spotřebováván cenný výkon. Otevřené GUI je v podstatě nečinné. Zpět k tlačítku *Start*. V jeho funkci je timer. Jedině zde se vytvoří.



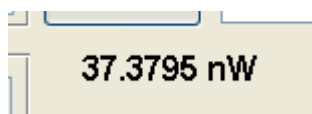
Obrázek č. 17: Vzhled editovatelného pole s periodou

Editovatelné pole udává periodu měření. Lze ji měnit před i v průběhu spuštěného timeru. Není třeba se periodou zabývat, až do chvíle, kdy ji chceme změnit. Minimální perioda je 1/6 sekundy, což je 166 milisekund. Proto při pokusu o nastavení nižší frekvence se nastaví 166 milisekund. Co se týče maximální periody, ta omezena není.



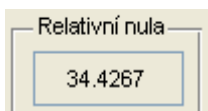
Obrázek č. 18: Vzhled editovatelného pole s počtem vzorků

Toto pole udává počet vzorků, které se mají vykonat. Implicitně je nastaveno nekonečně mnoho a nápis je „p. vzorku“. Při jakékoliv numerické hodnotě, zadané i během měření, se při každém cyklu odečte 1 a zbytek se zobrazí. Když počet vzorků dojde na nulu, přestane se měřit. Pokud uživatel opět spustí start a v poli je číslo nula, přepíše se na „inf“, což je nekonečno. Kdykoli během měření se hodnota může přepsat. Při zapsání jakéhokoli písmenka se počet vzorků nastaví na „inf“, což je nekonečno.



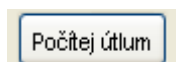
Obrázek č. 19: Vzhled textového pole s výkonem

V textovém poli je zobrazena aktuální hodnota optického výkonu. Automaticky se mění jednotky od nanowaty až po miliwaty. Hodnota má přesnost čtyř desetinných míst. Přesnost je stejná, jaká je odesílána měřicím přístrojem.



Obrázek č. 20: Vzhled editovatelného pole s relativní nulou

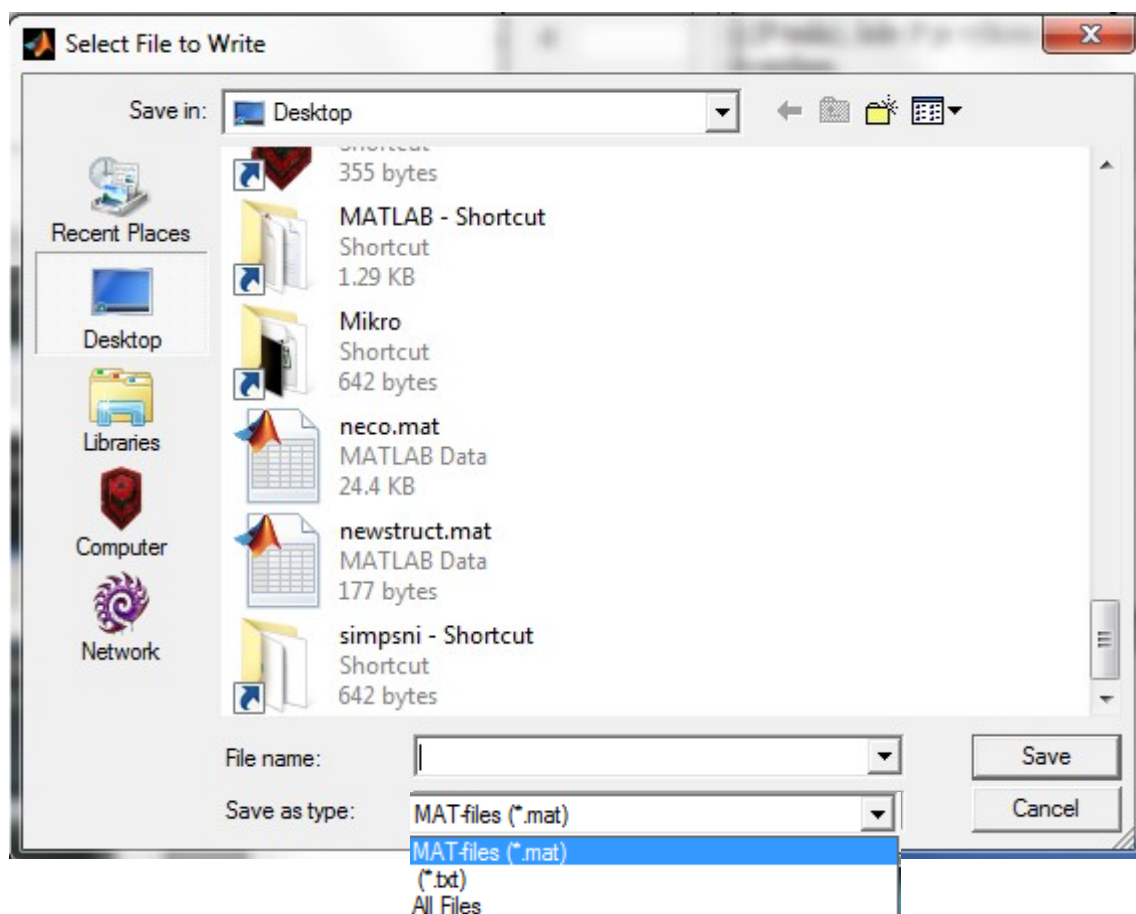
Editovatelné pole udává relativní nulu pro výpočet útlumu. Je vždy v nanowatech. Implicitně je nastavena nula, ale není nastaveno počítání útlumu. Při stisku tlačítka pro počítání útlumu se načte aktuální hodnota výkonu. Může však být kdykoli ručně přepsána.



Obrázek č. 21: Vzhled tlačítka pro počítání útlumu

Zamáčknutím tlačítka *Počítej útlum*, se okamžitě spočítá útlum z posledního výkonu a relativní nuly. Při každém dalším měření se už útlum počítá automaticky. Vzorec pro výpočet útlumu je roven  $10 \cdot \log(P/nula)$ , kde P je výkon naměřený měřicím přístrojem a nula, jak již bylo zmíněno, je zadána uživatelem.

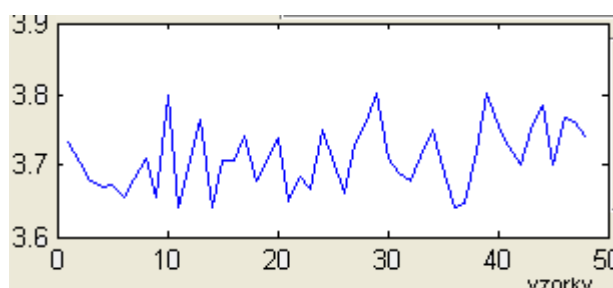
Při stisku tlačítka **Export** vyskočí nové okno pro výběr souboru. V případě, že soubor není vytvořen - vytvoří se, v případě že je - přepíše se. Jak již bylo zmíněno v kapitole 3.02 jsou dva druhy formátu. Uživatel si je nevybírám přímo. Vybírá si je druhem vybraného souboru – mat nebo jiným. Okno pro výběr souboru nám automaticky nabízí mat soubory. Jako druhá volba je .txt a jako třetí jsou všechny soubory.



Obrázek č. 22: Vzhled okna pro volbu souboru

Okno pro výběr exportu je závislé na operačním systému obslužného počítače. V tomto případě se jedná o Windows 7. Po výběru souboru tlačítko *Export* zůstává zakliknuté. Data se ukládají ve chvíli, kdy vyklikneme tlačítko *Export* a nebo při zavření GUI. To platí pro typ souboru mat. Pro jiné formáty se data ukládají průběžně. V případě, že si uživatel exportování rozmyslí a nevybere soubor, GUI se vrátí do stavu před stiskem tlačítka pro export. To znamená, že tlačítko exportu se vyklikne.

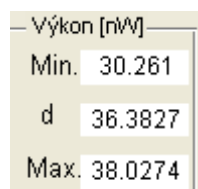
Tlačítko **Graf** po stisku začne vykreslovat průběh výkonu v závislosti na vzorcích do grafu. Po vykliknutí tlačítka grafu se graf nemaže a při následném stisku se vykresluje dále. Po stu vzorcích se první dva smažou. Takto vypadá ukázka grafu:



Obrázek č. 23: Vzhled grafu pro naměřený výkon

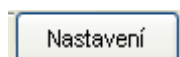
Tlačítko **RST** smaže graf.

Dalším segmentem GUI je výpis minimální, maximální a průměrné hodnoty měření:



Obrázek č. 24: Vzhled segmentu s hodnotami

Výkon je udáván v nanowatech a průměrný výkon je počítán pomocí vzorce:  $((\text{pocet\_prumer} * \text{prumer}) + \text{out}) / (\text{pocet\_prumer} + 1)$ , kde pocet\_prumer je pomocná proměnná udávající počet výkonů, vynásobí se dosavadním průměrem a přičte se aktuální výkon. To se vydělí počtem výkonů zvýšených o 1. Tento vzorec má výhodu, že složitost nezáleží na počtu odměřených vzorků.



Obrázek č. 25: Vzhled tlačítka pro tvorbu nastavení

Posledním částí první části aplikace je tlačítko *Nastavení*. Po jeho kliknutí je otevřeno nové okno aplikace. Jako vstup je jedna proměnná - sériové spojení.

The screenshot shows a software window titled 'prikazy'. It is divided into two main sections:

- Přikazy standardu IEEE488.2:** This section contains a list of standard commands, each with a 'Zadej' (Enter) button and an input field:
  - \*IDN? (Zadej)
  - \*TST? (Zadej) [0]
  - \*OPC (Zadej)
  - \*OPC? (Zadej)
  - \*WAI (Zadej)
  - \*RST (Zadej)
  - \*SRE (Numeric Zadej)
  - \*SRE? (Zadej) [0]
  - \*STB? (Zadej)
  - \*ESE (Numeric Zadej)
  - \*ESE? (Zadej) [0]
  - \*ESR? (Zadej)
  - \*CLS (Zadej)
- Specifické přikazy PM100:** This section contains specific commands for the PM100 device, each with a 'Zadej' button and an input field:
  - :POWER? (Zadej) [2.6881e-010]
  - :HEAD:INFO? (Zadej)
  - :WAVELENGTH? (Zadej)
  - :WAVELENGTH (Numeric Zadej)
  - :WAVELENGTH:RANGE? (Zadej)
  - :FILTER? (Zadej) [50HZ]
  - :FILTER (50HZ|60HZ Zadej)
  - :PHOTOCURRENT? (Zadej)
  - :PHOTOVOLTAGE? (Zadej) [0]
  - :DARKCURRENT? (Zadej) [0]
  - :DARKCURRENT (Numeric Zadej)
  - :DARKVOLTAGE? (Zadej) [0]
  - :DARKVOLTAGE (Numeric Zadej)
  - :ATTENUATION? (Zadej)
  - :ATTENUATION (Numeric Zadej)

Obrázek č. 26: Vzhled druhé části GUI

Okno tvoří veškeré příkazy pro měřicí přístroj. Ty s odpověďmi mají vpravo od tlačítka *Zadej* místo na odpověď. Příkazy, které mají jako vstup nějaký parametr, mají prepisovatelné pole. Samotným prepisem pole se vykoná příkaz. Originál aplikace od Thorlabs tuto část vůbec nemá. Uživatel by měl vědět, co dělá v této části programu, neboť špatnými zásahy může negativně ovlivnit měření - od filtrování špatné frekvence napájecího napětí (50 nebo 60 Hz), po nastavení napětí, nebo proudu naprázdno na špatný typ senzoru, až po relativně neškodné zesílení měřeného výsledku. Není však cesta, jak takovým chybám zabránit, proto by měl uživatel dobře zvážit, co se chystá zadat.

Příkazy standardu IEEE488.2

|       |         |       |  |
|-------|---------|-------|--|
| *IDN? |         | Zadej |  |
|       |         |       |  |
| *TST? |         | Zadej |  |
| *OPC  |         | Zadej |  |
| *OPC? |         | Zadej |  |
| *WAI  |         | Zadej |  |
| *RST  |         | Zadej |  |
| *SRE  | Numeric | Zadej |  |
| *SRE? |         | Zadej |  |
| *STB? |         | Zadej |  |
| *ESE  | Numeric | Zadej |  |
| *ESE? |         | Zadej |  |
| *ESR? |         | Zadej |  |
| *CLS  |         | Zadej |  |

Obrázek č. 27: První skupina příkazů

První část tvoří příkazy ze standardu IEEE488.2. V první části GUI se používá jen první příkaz této skupiny. Tyto příkazy vesměs odlaďují a zjišťují případné chyby. Například příkaz OPC udává, zda v bufferu nedošlo k přetečení řady odpovědí. To je zbytečné zjišťovat, neboť první částí GUI se nedá přístroj zahltit dotazy, jelikož je po každém dotazu okamžitě čtena odpověď. Ostatní příkazy jsou podobné. Jejich hodnoty a nastavení sice nejsou vždy zbytečné, ale značně by komplikovaly celé GUI, což je špatně. Proto jsou zde vypsány ve tvaru, jak je přijímá přístroj a je na uživateli, jak s nimi naloží. Opět opakuji, že tato druhá část GUI není pro laiky a bez případných manuálů je téměř nepoužitelná.

Specifické příkazy PM100

|  |  |                                      |
|--|--|--------------------------------------|
| :POWER?                                | <input type="button" value="Zadej"/>   | 4.59504e-012                         |
| :HEAD:INFO?                            | <input type="button" value="Zadej"/>   |                                      |
| 08100620, S120UV, D 50mW, 5.000000e-02 |  |                                      |
| :WAVELENGTH?                           | <input type="button" value="Zadej"/>   | 6.5e-007                             |
| :WAVELENGTH                            | <input type="text" value="Numeric"/>   | <input type="button" value="Zadej"/> |
| :WAVELENGTH:RANGE?                     | <input type="button" value="Zadej"/>   | 2.000000e-07,<br>1.100000e-06        |
| :FILTER?                               | <input type="button" value="Zadej"/>   | 50HZ                                 |
| :FILTER                                | <input type="text" value="50HZ 60HZ"/> | <input type="button" value="Zadej"/> |
| :PHOTOCURRENT?                         | <input type="button" value="Zadej"/>   | 1.76086e-011                         |
| :PHOTOVOLTAGE?                         | <input type="button" value="Zadej"/>   | 0                                    |
| :DARKCURRENT?                          | <input type="button" value="Zadej"/>   | 0                                    |
| :DARKCURRENT                           | <input type="text" value="Numeric"/>   | <input type="button" value="Zadej"/> |
| :DARKVOLTAGE?                          | <input type="button" value="Zadej"/>   | 0                                    |
| :DARKVOLTAGE                           | <input type="text" value="Numeric"/>   | <input type="button" value="Zadej"/> |
| :ATTENUATION?                          | <input type="button" value="Zadej"/>   | 0                                    |
| :ATTENUATION                           | <input type="text" value="Numeric"/>   | <input type="button" value="Zadej"/> |

Obrázek č. 28: Druhá skupina příkazů

Druhá část GUI obsahuje specifické příkazy měřicího přístroje. Prvních pět příkazů je použito v první části GUI. Ostatní příkazy vesměs slouží při nastavování nových senzorů a přídatných čoček.

## 4 Závěr

Cílem práce bylo zprovoznění komunikace v prostředí MATLAB pro měřicí přístroj PM100 a tvorba aplikace s graficko-uživatelským rozhraním pro práci s ním. GUI mělo být jednoduché a intuitivní.

Zprovoznění komunikace přes RS-232 se mi v MATLAB povedlo vytvořit. První část aplikace grafického rozhraní dovoluje uživateli ovládat měřicí přístroj a provádět různá nastavení. Optický výkon lze exportovat do dvou formátů. Tím hlavním je matice, která může být snadno načtena v MATLAB, kde lze výsledky dále zpracovávat. Druhým formátem exportu je standardizovaný formát csv. Druhá část aplikace obsahuje všechny příkazy, které jsou v manuálu měřicího přístroje definovány. Tato část není určena pro neznalé uživatele, práce probíhá přímo s příkazy.

Na mou bakalářskou práci navazuje diplomová práce s názvem „Možnosti měření profilu optického svazku“, která využívá zprovozněné sériové komunikace v MATLAB.



## 5 Seznam literatury:

- [1] ZAPLATÍLEK, K.; DOŇAR, B. [i]MATLAB: tvorba uživatelských aplikací.[/i] Vydání 1. Praha, BEN - technická literatura, 2004. 215 s. ISBN 80-7300-133-0.
- [2] *Www.thorlabs.de* [online]. 04-16-2006 [cit. 2011-03-10]. Datasheet senzoru S120B. Dostupné z WWW: <<http://www.thorlabs.de/Thorcat/13300/13336-S01.pdf>>.
- [3] *Www.thorlabs.de* [online]. 2006 [cit. 2011-03-10]. Operační manuál PM100I. Dostupné z WWW: <<http://www.thorlabs.de/Thorcat/12200/12271-D02.pdf>>.
- [4] *Www.mathworks.com* [online]. 2011 [cit. 2011-02-10]. Webové stránky firmy MathWorks. Dostupné z WWW: <<http://www.mathworks.com>>.
- [5] *Www.humusoft.cz* [online]. © HUMUSOFT 1991 - 2011 [cit. 2011-02-10]. Webové stránky obchodní společnosti HUMUSOFT s.r.o. . Dostupné z WWW: <<http://www.humusoft.cz/produkty/matlab/matlab/>>.
- [6] Rs232. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, , last modified on 11-04-2011 [cit. 2011-04-26]. Dostupné z WWW: <<http://en.wikipedia.org/wiki/Rs232>>.
- [7] *Www.arcelect.com* [online]. 2010 [cit. 2011-02-10]. Informace o sériovém portu RS232 Dostupné z WWW:< <http://www.arcelect.com/rs232.htm>>.
- [8] *Www.beyondlogic.org* [online]. 2010 [cit. 2011-04-26]. Rozhraní RS232. Dostupné z WWW: <<http://www.beyondlogic.org/serial/serial.htm>>.

## **Příloha**

Zdrojový kód programu je uložen v souborech final.m a prikazy.m, grafická část GUI je uložena v souborech final.fig a prikazy.fig. Všechny soubory tvoří přílohu tištěné verze této práce a jsou umístěny na samostatném nosiči CD ROM, kde jsou dále umístěny ukázky exportů, datasheet k senzoru S120B a operační manuál PM100.

### **Seznam tabulek:**

- Tabulka č. 1 : Označení pinů [3]  
Tabulka č. 2: Přenosové rychlosti [6],[7]  
Tabulka č. 3.: Přenosové rychlosti v OS Linux a Windows [8]  
Tabulka č. 4: Emulační příkazy [3]  
Tabulka č. 5: Příkazy ze standardu IEEE [3]  
Tabulka č. 6: Specifické příkazy PM100 [3]

### **Seznam obrázků:**

- Obrázek č. 1: Konektor typu Male (samec) [7]  
Obrázek č. 2 : Konektor s číselným označením pinů [7]  
Obrázek č. 3: Propojení pinů mezi dvěma konci RS-232 [3]  
Obrázek č. 4: Měřicí přístroj PM100 [3]  
Obrázek č. 5: Měřicí přístroj PM100 pohled shor [3]  
Obrázek č. 6: Měřicí přístroj PM100 pohled ze spodu [3]  
Obrázek č. 7: Optický senzor S120B [2]  
Obrázek č. 8: Vzhled aplikace pro PM100 od Thorlabs  
Obrázek č. 9: Hlášení komunikačního eroru  
Obrázek č. 10: vzhled GUI po spuštění  
Obrázek č. 11: Vzhled GUI během měření  
Obrázek č. 12: Vzhled volby COM a tlačítka *Připoj*  
Obrázek č. 13: Vzhled GUI během měření  
Obrázek č. 14: Vzhled políčka pro nastavení vlnové délky  
Obrázek č. 15: Vzhled části GUI s měřicími elementy  
Obrázek č. 16: Tlačítko start  
Obrázek č. 17: Vzhled editovatelného pole s periodou  
Obrázek č. 18: Vzhled editovatelného pole s počtem vzorků  
Obrázek č. 19: Vzhled textového pole s výkonem  
Obrázek č. 20: Vzhled editovatelného pole s relativní nlou  
Obrázek č. 21: Vzhled tlačítka pro počítání útlumu  
Obrázek č. 22: Vzhled okna pro volbu souboru  
Obrázek č. 23: Vzhled grafu pro naměřený výkon  
Obrázek č. 24: Vzhled segmentu s hodnotami  
Obrázek č. 25: Vzhled tlačítka pro tvorbu nastavení  
Obrázek č. 26: Vzhled druhé části GUI  
Obrázek č. 27: První skupina příkazů  
Obrázek č. 28: Druhá skupina příkazů

### **Seznam příloh:**

- final.m  
prikazy.m  
final.fig  
prikazy.fig  
datasheetS120B.pdf  
opracni\_manual\_PM100.pdf  
juc036\_FEI\_PM100\_2011.pdf  
ukázky exportů